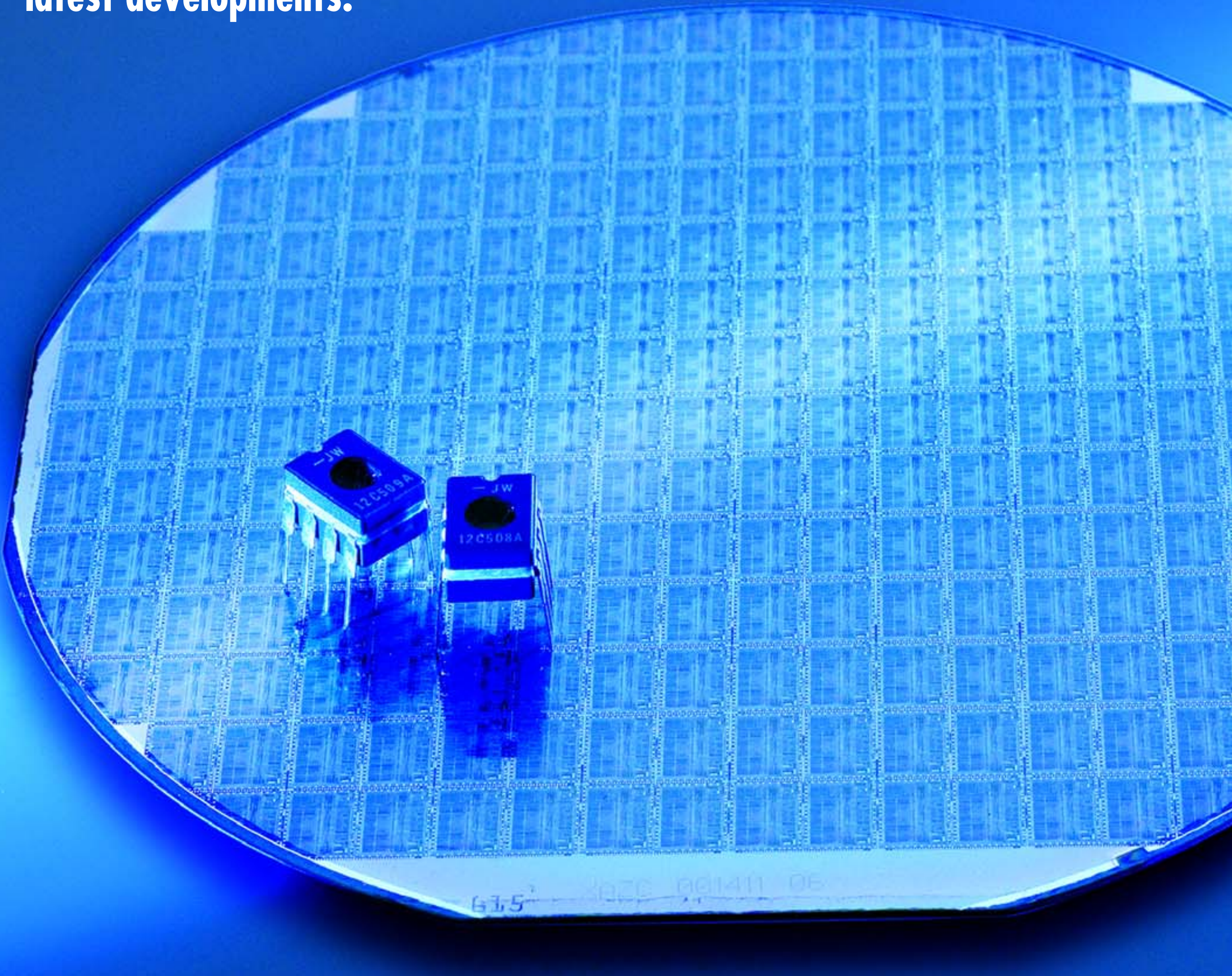
TRENDS IN MICROC

Christian Tavernier

The number of different microcontrollers on offer just keeps on growing. Nearly every single week a new product appears on the market. Choosing the right microcontroller for a particular design is becoming increasingly more difficult. Reason enough to appraise the latest developments.



ONTROLLERS

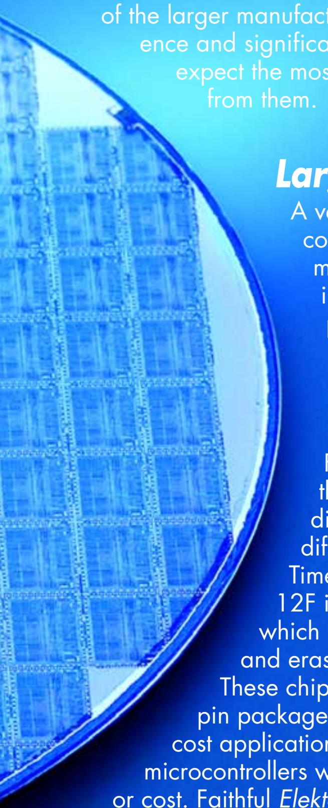
AN ABUNDANCE OF CHOICE

The microcontroller market is mostly dominated by a few ‘heavyweights’ such as Microchip, Atmel and, to a lesser extend, Philips. In addition, there are many smaller manufacturers who make very interesting developments in this area. There are also a number of manufacturers who are not at all that well known for their microcontroller products, and are often forgotten as a consequence. Names that come to mind are Toshiba with its TMP86xxx-family, and Zilog, the inventor of the famous Z80, who now offers the Z8 Encore! and eZ80 families. Also don’t forget Dallas Semiconductor with its DS89C420, Cypress with its PsoC (Programmable System-on-Chip) such as the CY8C27x, or the, until recently, completely unknown company Cygnal with its C8051xxx.

16-bit devices appear to be old hat

The majority of these products, however interesting, will have to have some very desirable properties in order to secure a prominent spot in an already oversupplied market. A few years ago, the PIC from Microchip succeeded in doing this. For other manufacturers this is not a problem; Toshiba, for instance, makes enough end products such as notebooks and other consumer electronics to use plenty of its own microcontrollers.

In this article we will concentrate mostly on the products of the larger manufacturers, because their broad experience and significant head start means that we can expect the most innovative developments to come from them.



Larger or smaller?

A very obvious trend is that the microcontrollers are becoming increasingly more powerful: more memory, more inputs and outputs, more integrated peripherals, etc. A lesser-known development is currently in progress at the bottom end of the market, both at Microchip, and, to a smaller extend, at Atmel.

For example, Microchip developed the 12Fxxx from the 12Cxxx. The different letter indicates an important difference: the 12C is an OTP (One Time Programmable) device, while the 12F is provided with Flash memory, which can be electronically programmed and erased a few thousand times.

These chips are available in either 8- or 14-pin packages and are mainly intended for low cost applications where until a few years ago microcontrollers were avoided because of their size or cost. Faithful *Elektor Electronics* readers will know the 12C508 (**Figure 1**), the first member of this family. But let us first take a look at the 12F629, which, in its 8-pin package has the following to offer: 1 k Flash program memory, an EEPROM 128 bytes in size, 64 bytes of RAM, 6 I/O-lines, an 8-bit and a 16-bit programmable timer. If all of this is not enough, the 12F675 may be more appropriate. Incidentally, the numbering scheme escapes us. This one is identical to the 629, but in addition provides a 10-bit, 4-channel ADC. These ICs operate from an internal RC clock source up to 4 MHz or from an external crystal up to 20 MHz.

Atmel hasn’t been sitting idle in this area either, and has the ATtiny-family, which comprises nine different versions. Even though it can’t really be called new, the ATtiny2313 deserves some of our attention. It has a 2 k Flash memory, a UART (Universal Asynchronous Receiver/Transmitter) and up to 18 I/O-lines. The crystal frequency may be up to 20 MHz. The controller is housed in a 20-pin package.

The rfPIC or ‘radio’ PIC

In the area of ‘small’ microcontrollers it is Microchip who steals the show.

Now that microcontrollers are becoming more common in remote controls, Microchip has decided to offer the rfPIC, literally the radio frequency PIC. This one is part of the small PIC-family that we discussed just before. In a single package, the rfPIC contains, in addition to the microcontroller, a complete circuit for an ASK- or FSK-transmitter (Amplitude or Frequency Shift Keying). Also provided, to ensure a stable frequency, a VCO (Voltage Controlled Oscillator) and a PLL (Phase Locked Loop), based on the crystal frequency of the circuit.

The basic version, the rf12C509 comprises a 12C509 and a built-in transmitter. But for the creative among us the rf12F675 is much more interesting. This is, after all, the aforementioned 12F675 with the radio added. Applying these controllers is very simple, as is illustrated by **Figure 2**.

Trends

Beside the usual increase of on-chip, integrated memory, the last few years show two more main areas of microcontroller development.

The first concerns the operating speed. Although countless circuits are still provided with a 4-MHz crystal, the product catalogues from the Microchip, Atmel and Motorola are full with ICs that will function at 16 or 20 MHz. Note that it is not that straightforward to just compare clock speeds. An ATmega128 from Atmel operates on a clock frequency of ‘only’ 16 MHz, but executes nearly all instructions in a single clock cycle. On the other hand, an MC9S12D from the HCS12 family from Motorola runs at 25 MHz but requires multiple clock cycles per instruction.

The second main trend concerns the specialisation of the internal peripherals. The UART and SPI interfaces have been common in the higher-end microcontrollers from most manufacturers for many years. But now more specialised interfaces are being added. Virtually all manu-

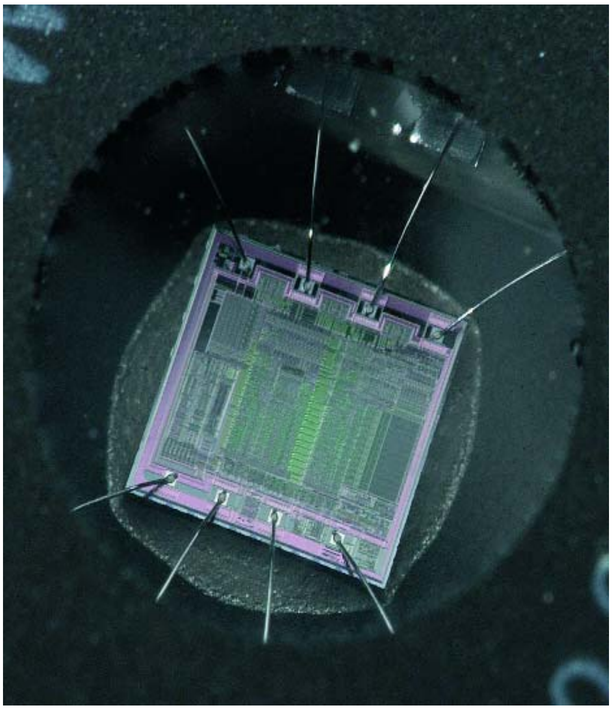


Figure 1. A window through which the memory is erased with ultraviolet radiation. This image will soon be condemned to the past as a consequence of the rise of Flash memory.

facturers support the I^{2}C bus. Sometimes the interface operates only in the simplest slave mode, but more often than not, in master mode as well. The USB (Universal Serial Bus) interface, popular because of the PC-market, is quickly making its entry into the microcontroller world. Microchip has the 16C745, the 765 and the future 18F2455, 255, etc. Atmel offers the AT91RM3400 and even Motorola has adapted a version of the old 6805, such as the 68HC705JBx.

The use of microcontrollers in cars is becoming increasingly important

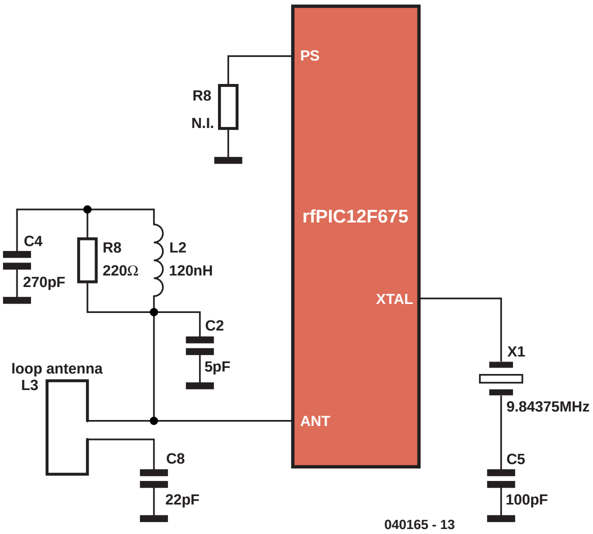


Figure 2. The RF part in an rfPIC is easy to use.

ARM

On our whirlwind tour of the microcontroller world we cannot ignore the products from ARM. Although ARM, strictly speaking, doesn't actually manufacture microcontrollers, it has provided for many years the 32-bit cores that well known manufacturers such as AMI, Atmel, Cirrus, Philips, Samsung, STMicroelectronics, Texas Instruments and even Intel happily use in their own products. We get the impression that this concerns an important development. Why worry about developing your own 32-bit microcontroller when an industry-accepted core already exists? The need for increasingly advanced products, which, at the same time, are still easy to maintain and use, inevitably influences the intelligence of each system: the microcontroller. The need for an ever-increasing number of functions translates into a transition from the traditional 4- or 8-bit controllers to 32-bits. It seems that progress has skipped over the 16-bit versions, because 32-bit offers a much higher performance and unequalled flexibility.

Applications

Especially in cars, the use of microcontrollers is becoming increasingly important. This has resulted in manufacturers not only integrating CAN-bus controllers on their chips but the microcontrollers themselves are becoming more advanced. This is the case, for example, with the 18F2332 from Microchip, which can operate at 40 MHz. The same is true for the MC68HC908MRx from Motorola.

Even in this ocean of digital chips, the analogue world has not been forgotten. These days, all manufacturers have ICs in their product catalogues with advanced analogue/digital-converters. An example is the ATmega128 from Atmel. It contains a 10-bit, 8-channel model, which operates in normal mode (single ended) as an 8-channel converter. 7-channel differential mode is also possible and 2 channels are provided with an integrated programmable amplifier (1 to 200 times). For many applications there is no longer the need to resort to an external converter.

The addition of internal peripherals has no negative impact on the amount of memory. On the contrary, the better-endowed ICs often contain memories of amazing size. **Table 1** provides an overview of the three 'largest' presently available microcontrollers from Microchip (the 18Cxxx-family), from Atmel (the ATmega-family) and from Motorola (the HCS12-family). Conspicuous in this table is the MCS12H256B from Motorola (**Figure 3**) with the overwhelming 99 parallel I/O-lines and 256K-Flash memory. This is quite a bit different than the 68705P3 with 20 I/O-lines and 1.6K-memory.

Phoenix

You know, the mythical bird that rose from its own ashes. There is a microcontroller that we certainly have to give some attention despite the fact that we're now in the second half of 2004: the 8051. Even though this microcon-

troller, in its basic form, first appeared on the market at the end of the 80's, it is still the basis for many new developments.
Despite its respectable age, the 8051 has survived thanks not only to the addition of new internal peripherals and expansions to the program memory, but also additional features that used to be considered impossible (such as in-circuit-programming, discussed further on in this article).

If the 8051 assembler doesn't hold any secrets for you any more, then you should take a look at the AT89xxx family from Atmel. All ICs in this family were developed from the 8051, but have in-circuit-programming, new internal peripherals and expanded program memories. If you find that these ICs are a little too 'well-done' then maybe the new Philips family P89LPC900 has something to offer you. If this name doesn't mean anything to you then you should dive into your *Elektor Electronics* archive for the October and November 2003 issues in which this IC was discussed extensively.

Programs and development tools

Initially, that is to say, when a microcontroller barely contained 1 or 2 kB of memory, programming in assembler was the only real option. The available memory would easily have been filled with a few lines of a higher programming language, because a single line usually results in numerous lines of assembly language as a result of the 'expansion urge' of the compiler.

The assembler is still an indispensable tool if we want to create a genuine real-time program or if we want to obtain the greatest possible performance out of the microcontroller. But these days, the preference is to use a higher programming language. Most programmers now select either BASIC or C in preference to assembly.
There are now C compilers for all microcontroller families. This varies from the quite pricey products such as the

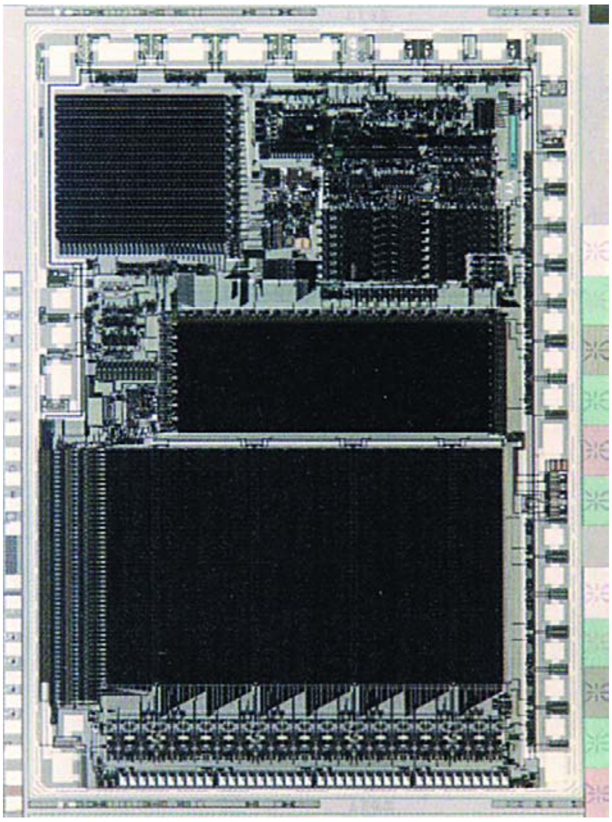
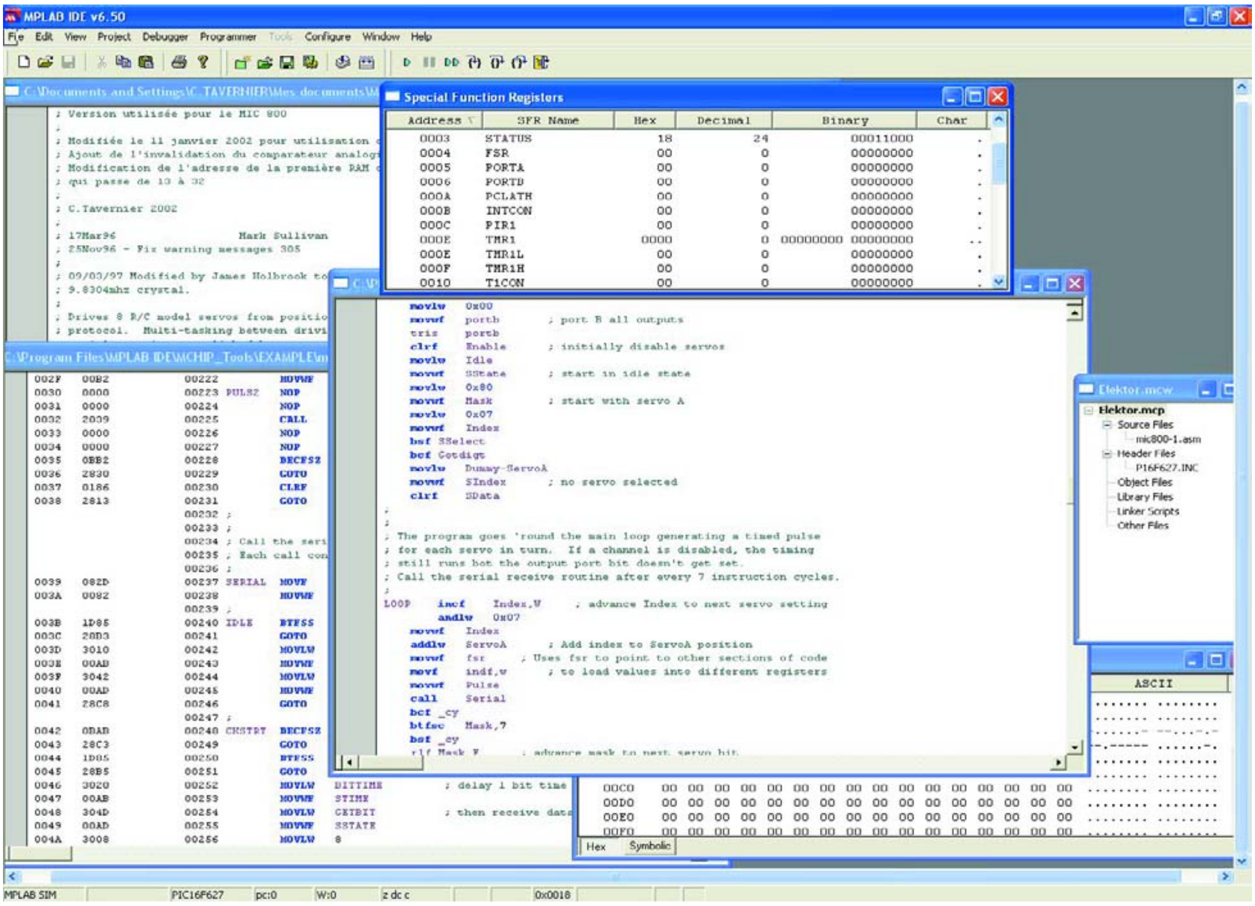


Figure 3. Photograph of a microcontroller-chip.

Table 1. Main specs of three modern controllers from Atmel, Microchip and Motorola.

Parameter	**ATmega128 (Atmel)**	**18F8720 (Microchip)**	**MC9S12H256 (Motorola)**
Flash program memory	128 K	128 K	256 K
EEPROM data memory	4 K	1 K	4 K
RAM memory	4 K	3.8 K	12 K
8-bit timers	2	2	0
16-bit timers	2	3	1
Capture and compare timers	4	5	8
PWM channels	8	5	6
I/O lines	53	68	99
UART (serial asynchronous interface)	2	2	2
SPI (serial synchronous interface)	1	1	1
I^{2}C interface	–	1	1
A/D converter	8 channels	16 channels	16 channels
A/D resolution	10 bits	10 bits	10 bits
Analogue comparators	1	2	-
Maximum clock frequency	16 MHz	40 MHz	32 MHz


MPLAB IDE v6.50
File Edit View Project Debugger Programmer Tools Configure Window Help
C:\Documents and Settings\C.TAVERNIER\Mes documents\M
C:\Program Files\MPLAB IDE\MCHIP_Tools\EXAMPLE\m
Elektor.mcw
Hex
Symbolic
MPLAB SIM
PIC16F627
pc:0
W:0
z dc c
0x0018

```
; Version utilisée pour le MIC 800
;
; Modifiée le 11 janvier 2002 pour utilisation c
; Ajout de l'invalidation du comparateur analogi
; Modification de l'adresse de la première RAM c
; qui passe de 13 à 32
;
; C.Tavernier 2002
;
; 17Mar96                         Mark Sullivan
; 25Nov96 - Fix warning messages 305
;
; 09/03/97 Modified by James Holbrook to
; 9.8304mhz crystal.
;
; Drives 8 R/C model servos from positio
; protocol.  Multi-tasking between drivi
```

Special Function Registers

Address	SFR Name	Hex	Decimal	Binary	Char
0003	STATUS	18	24	00011000	.
0004	FSR	00	0	00000000	.
0005	PORTA	00	0	00000000	.
0006	PORTB	00	0	00000000	.
000A	PCLATH	00	0	00000000	.
000B	INTCON	00	0	00000000	.
000C	PIR1	00	0	00000000	.
000E	TMR1	0000	0	00000000 00000000	..
000E	TMR1L	00	0	00000000	.
000F	TMR1H	00	0	00000000	.
0010	T1CON	00	0	00000000	.

```
0029    00B2           00222         MOVWF
0030    0000           00223 PULS2   NOP
0031    0000           00224         NOP
0032    2039           00225         CALL
0033    0000           00226         NOP
0034    0000           00227         NOP
0035    0BB2           00228         DECFSZ
0036    2830           00229         GOTO
0037    0186           00230         CLRF
0038    2813           00231         GOTO
                       00232 ;
                       00233 ;
                       00234 ; Call the seri
                       00235 ; Each call con
                       00236 ;
0039    082D           00237 SERIAL  MOVF
003A    0082           00238         MOVWF
                       00239 ;
003B    1D85           00240 IDLE    BTFSS
003C    28D3           00241         GOTO
003D    3010           00242         MOVLW
003E    00AD           00243         MOVWF
003F    3042           00244         MOVLW
0040    00AD           00245         MOVWF
0041    28C8           00246         GOTO
                       00247 ;
0042    0BAD           00248 CHSTRT  DECFSZ
0043    28C3           00249         GOTO
0044    1D05           00250         BTFSS
0045    28B5           00251         GOTO
0046    3020           00252         MOVLW   BITTIME        ; delay 1 bit time
0047    00AB           00253         MOVWF   STIME
0048    304D           00254         MOVLW   GETBIT         ; then receive data
0049    00AD           00255         MOVWF   SSTATE
004A    3008           00256         MOVLW   8
```

```
    movlw   0x00
    movwf   portb       ; port B all outputs
    tris    portb
    clrf    Enable      ; initially disable servos
    movlw   Idle
    movwf   SState      ; start in idle state
    movlw   0x80
    movwf   Mask        ; start with servo A
    movlw   0x07
    movwf   Index
    bsf SSelect
    bcf Gotdigt
    movlw   Dummy-ServoA
    movwf   SIndex      ; no servo selected
    clrf    SData
;
;
; The program goes 'round the main loop generating a timed pulse
; for each servo in turn.  If a channel is disabled, the timing
; still runs bot the output port bit doesn't get set.
; Call the serial receive routine after every 7 instruction cycles.
;
LOOP    incf    Index,W     ; advance Index to next servo setting
        andlw   0x07
    movwf   Index
    addlw   ServoA      ; Add index to ServoA position
    movwf   fsr     ; Uses fsr to point to other sections of code
    movf    indf,w      ; to load values into different registers
    movwf   Pulse
    call    Serial
    bcf _cy
    btfsc   Mask,7
    bsf _cy
    rlf Mask F      ; advance mask to next servo bit
```

```
Elektor.mcp
  Source Files
    mic800-1.asm
  Header Files
    P16F627.INC
  Object Files
  Library Files
  Linker Scripts
  Other Files
```

```
00C0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ........ ........
00D0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ........ ........
00E0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ........ ........
00F0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ........ ........
```

Figure 4. Although MPLAB is free, it contains everything to develop an application for a PIC with a maximum of user friendliness.

CCS C PIC compiler to the freeware PICC-Lite C compiler from Htsoft, which is often sufficient for many applications.

For those who prefer not to work in C or assembler there are excellent BASIC-compilers, such as the PicBasic Pro Compiler from Micro Engineering Labs and Proton from Crownhill Associates. For the AVR-family from Atmel you could use Bascom AVR from MCS Electronics.

Most programmers now prefer BASIC or C over assembler

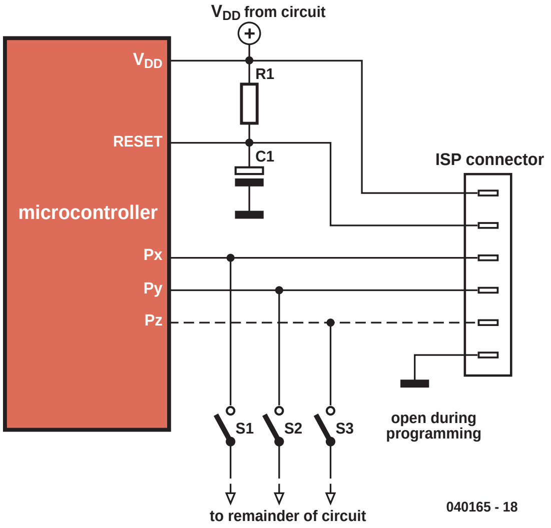


Figure 5. Operating principle of in-circuit programming (ISP or ICSP).

For the 8051 family there is, among others, a special Bascom version from MCS, and SDCC, a free C compiler. Only the Motorola products suffer from a sparse availability of compilers, since the amount on offer is absolutely minimal.

Accepting the fact that most software developers often use identical programming elements, Microchip has come up with an original idea: a new program called Maestro. Using this, you can develop an application with next to no actual programming. This development environment can be downloaded free from the Microchip website and makes it possible to stick together standard modules. In this way a program can be put together nearly automatically.

In addition to a compiler, a microcontroller project also requires other development tools. Microchip and Atmel have the upper hand here with MPLAB (**Figure 4**) and AVR Studio respectively. These development environments, running under Windows, contain everything a programmer could only dream of a few years ago, including a program simulator. This development is still in progress, because MPLAB is currently up to version 6.5 and AVR Studio is up to version 4, so that even new products are well supported.

Despite the remarkable number of features these development tools are completely free. You only need to download them from the manufacturer's website to get started. It has to be noted that Motorola and other lesser-known manufacturers are running behind in this area, certainly when compared with these free products.

In circuit programming

Although it isn't really new, ISP (In System Programming) or ICSP (In Circuit Serial Programming) is without doubt the most important development regarding the programming aspect. When the programmer has created the program in the correct form, microcontrollers that support it can be programmed from a standard PC via a very simple interface, while the controller itself can remain in the circuit.
Figure 5 illustrates the operating principle of this concept. The microcontroller obtains its power supply and clock signal from the PCB. Two or three port pins on the IC temporarily have an alternative function, which makes it possible to erase and program the program memory. If the application doesn't use these pins, then they may be connected directly to a PC. Otherwise a few jumpers or DIP switches may be required to isolate the circuit during programming.

> Free development tools running under Windows contain everything that programmers until recently could only dream of

All modern microcontrollers with Flash memory support this programming method and on the Internet countless free software can be found for all the various controllers to program them in this way.
A program that distinguishes itself in this respect is FLIP (**Figure 6**). It is offered free by Atmel and can program countless microcontrollers from this manufacturer, provided they have some kind of serial interface (RS232, SPI, USB and even CAN-bus). This is currently the most versatile program around.

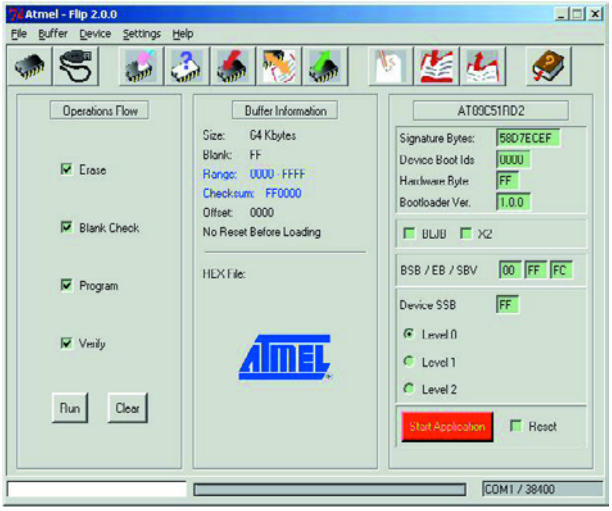
Microcontrollers or not?

In this article we must not forget to mention special microcontrollers such as the Basic Stamp, the PIC Basic and the Basic Tiger. These ICs with 24, 28 or 40 pins have, on a tiny PCB, a fast microcontroller, which is pre-programmed with an interpreter for a higher language. This is usually BASIC, but other programming languages are emerging, such as Java for the Javelin Stamp from Parallax. Even programming in an object-oriented language is possible with the OOPic.
Even though these products are very successful, they are very expensive compared to real microcontrollers, mainly because of they way they are manufactured. They seem to be most suitable for experimental applications or small production runs.

Conclusion

Much more can be said on this topic. But if we have to summarise the developments of the microcontroller area in a few words, then we can say that the advance of microcontrollers in all aspects of electronics is becoming increasingly important. These can be either more powerful devices on the one hand or with ever smaller and cheaper versions on the other hand. The application of microcontrollers is becoming easier and easier with free development tools and powerful compilers.
The microcontroller has become an indispensable part of electronics. This is testified by the many projects containing microcontrollers that have already appeared and will appear in *Elektor Electronics*.

(040165-1)



☑ Erase
☑ Blank Check
☑ Program
☑ Verify

Size: 64 Kbytes
Blank: FF
Range: 0000 - FFFF
Checksum: FF0000
Offset: 0000
No Reset Before Loading
HEX File:

Signature Bytes: 58D7ECEF
Device Boot Ids: 0000
Hardware Byte: FF
Bootloader Ver.: 1.0.0
☐ BLJB ☐ X2
BSB / EB / SBV: 00 FF FC
Device SSB: FF
● Level 0
○ Level 1
○ Level 2
☐ Reset
COM1 / 38400

Figure 6. The FLIP software program from Atmel for the in-circuit programming via practically any serial connection.

Internet addresses

Author's website (in French): http://www.tavernier-c.com/

Conventional microcontrollers and development tools

ARM:	http://www.arm.com/
Atmel:	http://www.atmel.com/
Microchip:	http://www.microchip.com/
Cygnal:	http://www.cygnal.com/
Cypress:	http://www.cypress.com/
Motorola:	http://mot-sps.com/
Philips:	http://www.semiconductors.philips.com/
Toshiba:	http://www.toshiba.com/taec/
Zilog:	http://www.zilog.com/

Special microcontrollers

Parallax:	http://www.parallax.com/
OOPic:	http://www.oopic.com/
Basic Tiger:	http://www.wilke.de/

Compilers

CCS C Pic Compiler:	http://www.ccsinfo.com/
PICC-Lite C Compiler:	http://www.htsoft.com/
Bascom AVR en Bascom 8051:	http://www.mcselec.com/
Proton - Basic compiler:	http://www.picbasic.org/
PicBasic (Pro) Compiler:	http://www.melabs.com/
SDCC Compiler:	http://sourceforge.net/projects/sdcc